We claim:

1. A method for protecting a network from a virus contained in an e-mail message as executable code, the method comprising:

(a) receiving the e-mail message in a gatekeeper server;

(b) converting the executable code from an executable format to a non-executable format; and

(c) forwarding the non-executable format to the recipient of the e-mail message.

2. The method of claim 1, wherein the executable code is contained in a body of the e-mail message.

3. The method of claim 2, wherein the executable code comprises a hypertext link, and wherein step (b) comprises deactivating the hypertext link.

4. The method of claim 1, wherein the executable code is contained in an attachment in the e-mail message.

5. The method of claim 4, wherein step (b) comprises:

(i) forwarding the attachment from the gatekeeper server to a sacrificial server; and

(ii) converting the attachment to the non-executable format on the sacrificial server.

6. The method of claim 5, wherein step (b) further comprises (iii) examining the sacrificial server for virus activity.

7. The method of claim 6, wherein step (b) further comprises (iv) rebooting the sacrificial server from a safe copy of an operating system obtained from a read-only device.

8. The method of claim 5, wherein communications between the gatekeeper server and the sacrificial server are authenticated using a challenge-and-response technique.

9. The method of claim 4, wherein step (b) comprises:

(i) maintaining a list of approved attachment types;

(ii) determining whether the attachment is of a type which is in the list of approved attachment types; and

(iii) if the attachment is not of a type which is in the list of approved attachment types, informing the recipient that a message containing a non-approved attachment has been received.

10. The method of claim 1, wherein step (b) comprises:

(i) maintaining a list of approved executable code;

(ii) determining whether the executable code is in the list of approved executable code; and

(iii) deactivating the executable code if the executable code is not in the list of approved executable code.

11. The method of claim 10, wherein:

the list of approved executable code includes information for determining whether the approved executable code has been altered; and

step (b) further comprises:

(iv) determining whether the executable code has been altered; and

(v) deactivating the executable code if the executable code has been altered.

12. The method of claim 11, wherein step (b)(iv) is performed through an algorithmic technique.

13. The method of claim 12, wherein the algorithmic technique is a check-summing technique.

14. The method of claim 12, wherein the algorithmic technique is a hashing technique.

15. The method of claim 1, wherein step (b) comprises:

(i) forming a first copy and a second copy of at least a portion of the e-mail message containing the executable code;

(ii) executing the executable code in the first copy but not the second copy; and

(iii) after the executable code in the first copy has been executed, comparing the first copy to the second copy to determine an effect of the executable code.

16. A system for protecting a network from a virus contained in an e-mail message as executable code, the system comprising:

a workstation computer on the network used by an recipient of the e-mail message;

a gatekeeper server, in communication with the workstation computer over the network, for receiving the e-mail message; and

a computer on the network for converting the executable code from an executable format to a non-executable format and forwarding the non-executable format to the workstation computer.

17. The system of claim 16, wherein the executable code is contained in a body of the e-mail message.

18. The system of claim 17, wherein the executable code comprises a hypertext link, and wherein the computer for converting deactivates the hypertext link.

19. The system of claim 16, wherein the executable code is contained in an attachment in the e-mail message.

20. The system of claim 16, wherein the computer for converting is a sacrificial server which is separate from the gatekeeper server.

21. The system of claim 20, wherein the sacrificial server is examined for virus activity.

22. The system of claim 21, wherein the network further comprises a read-only device, and wherein the sacrificial server is rebooted from a safe copy of an operating system obtained from the read-only device.

23. The system of claim 20, wherein communications between the gatekeeper server and the sacrificial server are authenticated using a challenge-and-response technique.

24. The system of claim 16, wherein the network maintains a list of approved attachment types, determines whether the attachment is of a type which is in the list of approved attachment types, and, if the attachment is not of a type which is in the list of approved attachment types, informs the recipient that a message containing a non-approved attachment has been received.

25. The system of claim 16, wherein the network maintains a list of approved executable code, determines whether the executable code is in the list of approved executable code, and deactivates the executable code if the executable code is not in the list of approved executable code.

26. The system of claim 25, wherein:

the list of approved executable code includes information for determining whether the approved executable code has been altered;

the network determines whether the executable code has been altered; and

the executable code is deactivated if the executable code has been altered.

27. The system of claim 26, wherein the system determines whether the executable code has been altered through an algorithmic technique.

28. The system of claim 27, wherein the algorithmic technique is a check-summing technique.

29. The system of claim 27, wherein the algorithmic technique is a hashing technique.

30. The system of claim 16, wherein the computer for converting converts the executable code by:

(i) forming a first copy and a second copy of at least a portion of the e-mail message containing the executable code;

(ii) executing the executable code in the first copy but not the second copy; and

(iii) after the executable code in the first copy has been executed, comparing the first copy to the second copy to determine an effect of the executable code.

Sub A6

31. A sacrificial server for use on a network, the sacrificial server comprising:

communication means for receiving an e-mail attachment from the network; and

processing means for converting the e-mail attachment from an executable format to a non-executable format and for returning the e-mail attachment to the network.

32. The sacrificial server of claim 31, wherein the sacrificial server is examined for virus activity.

33. The sacrificial server of claim 32, wherein the sacrificial server further comprises a read-only device and is rebooted from a safe copy of an operating system obtained from the read-only device.

34. The sacrificial server of claim 31, wherein communications between the network and the sacrificial server are authenticated using a challenge-and-response technique.

Sub A7

35. The sacrificial server of claim 31, wherein the sacrificial server stores a list of approved attachment types, determines whether the attachment is of a type which is in

the list of approved attachment types, and, if the attachment is not of a type which is in the list of approved attachment types, and informs the network that a message containing a non-approved attachment has been received.

36. The sacrificial server of claim 31, wherein the sacrificial server maintains a list of approved executable code, determines whether the attachment contains executable code and whether the executable code is in the list of approved executable code, and deactivates the executable code if the executable code is not in the list of approved executable code.

37. The sacrificial server of claim 36, wherein:

the list of approved executable code includes information for determining whether the approved executable code has been altered;

if the executable code is in the list of approved executable code, the sacrificial server determines whether the executable code has been altered; and

the executable code is deactivated if the executable code has been altered.

38. The sacrificial server of claim 32, wherein the sacrificial server determines whether the executable code has been altered through the use of an algorithmic technique

39. The sacrificial server of claim 38, wherein the algorithmic technique is a check-summing technique.

40. The sacrificial server of claim 38, wherein the algorithmic technique is a hashing technique.

41. The sacrificial server of claim 31, wherein the processing means converts the executable code by:

(i) forming a first copy and a second copy of at least a portion of the e-mail message containing the executable code;

(ii) executing the executable code in the first copy but not the second copy; and

(iii) after the executable code in the first copy has been executed, comparing the first copy to the second copy to determine an effect of the executable code.